Classification with SVMs

- Given a new point $x$, we can score it:
  - i.e., compute score: $w^T x + b = \sum a_i y_i x_i^T x + b$
  - Can set confidence threshold $t$.

Score $> t$ : yes
Score $< -t$ : no
Else: don’t know
Soft Margin Classification

- If the training data is not linearly separable, *slack variables* \( \xi_j \) can be added to allow misclassification of difficult or noisy examples.

- Allow some errors
  - Let some points be moved to where they belong, at a cost

- Still, try to minimize training set errors, and to place hyperplane “far” from each class (large margin)
Soft Margin Classification
Mathematically

- The old formulation:

\[
\begin{align*}
\text{Find } w \text{ and } b \text{ such that} \\
\Phi(w) &= \frac{1}{2} w^T w \text{ is minimized and for all } \{(x_i, y_i)\} \\
y_i (w^T x_i + b) &\geq 1
\end{align*}
\]

- The new formulation incorporating slack variables:

\[
\begin{align*}
\text{Find } w \text{ and } b \text{ such that} \\
\Phi(w) &= \frac{1}{2} w^T w + C \sum i \xi_i \text{ is minimized and for all } \{(x_i, y_i)\} \\
y_i (w^T x_i + b) &\geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for all } i
\end{align*}
\]

- Parameter $C$ can be viewed as a way to control overfitting – a regularization term
How fast are SVMs?

- **Training**
  - Time for training is dominated by the time for solving the underlying quadratic programming problem
  - Slower than Naïve Bayes
  - Non-linear SVMs are worse

- **Testing**
  - Fast - as long as we don’t have too many support vectors
What about more than 2 classes?

- SVMs are inherently 2 class classifiers
- Solution?
  - OVA
  - OVO
  - Multiclass SVMs
Linear SVMs: Summary

- The classifier is a *separating hyperplane*.
- Most “important” training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $x_i$ are support vectors with non-zero Lagrangian multipliers $\alpha_i$.
- **Both in the dual formulation of the problem and in the solution training points appear only inside inner products:**

Find $\alpha_1...\alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized and

1. $\sum \alpha_i y_i = 0$
2. $0 \leq \alpha_i \leq C$ for all $\alpha_i$

$f(x) = \sum \alpha_i y_i x_i^T x + b$
Non-linear SVMs

- Datasets that are linearly separable (with some noise) work out great:

- But what are we going to do if the dataset is just too hard?

- How about … mapping data to a higher-dimensional space:
Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:

$$\Phi: x \rightarrow \varphi(x)$$
The “Kernel Trick”

- The linear classifier relies on an inner product between vectors $K(x_i, x_j) = x_i^T x_j$

- If every datapoint is mapped into high-dimensional space via some transformation $\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$

- A kernel function is some function that corresponds to an dot product in some expanded feature space.

$$f(x) = \text{sign}(\sum \alpha_i y_i K(x_i, x) + b)$$
The “Kernel Trick”

Example:

- 2-dimensional vectors \( \mathbf{u} = [u_1, u_2] \) and \( \mathbf{v} = [v_1, v_2] \); let
  \[
  K(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u}^T \mathbf{v})^2,
  \]

- Need to show that \( K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^T \phi(\mathbf{v}) \):

- Consider \( \phi(\mathbf{u}) = [1 \ u_1^2 \sqrt{2} \ u_1 u_2 \ u_2^2 \ \sqrt{2} u_1 \ \sqrt{2} u_2] \)

\[
K(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u}^T \mathbf{v})^2 = (1 + u_1 v_1 + u_2 v_2)^2
\]

\[
= 1 + u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2 + 2 u_1 v_1 + 2 u_2 v_2
\]

\[
= [1 \ u_1^2 \sqrt{2} u_1 u_2 \ u_2^2 \ \sqrt{2} u_1 \ \sqrt{2} u_2]^T [1 \ v_1^2 \sqrt{2} v_1 v_2 \ v_2^2 \ \sqrt{2} v_1 \ \sqrt{2} v_2]
\]

\[
= \phi(\mathbf{u})^T \phi(\mathbf{v})
\]
Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial $K(x,z) = (1+x^Tz)^d$
    - Gives feature conjunctions
  - Radial basis function (infinite dimensional space)

$$K(x_i, x_j) = e^{-\|x_i-x_j\|^2/2\sigma^2}$$
Evaluation: Classic Reuters-21578 Data Set

- Most (over)used data set
- 21578 documents
- 9603 training, 3299 test articles (ModApte split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions (118 2-class classifiers)
- Average number of classes assigned
  - 1.24 for docs with at least one category
- Only about 10 out of 118 categories are large

Common categories (#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369, 119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)
The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter
Per class evaluation measures

- **Recall**: Fraction of docs in class \(i\) classified correctly:
  \[
  \frac{c_{ii}}{\sum_j c_{ij}}
  \]

- **Precision**: Fraction of docs assigned class \(i\) that are actually about class \(i\):
  \[
  \frac{c_{ii}}{\sum_j c_{ji}}
  \]

- **F Measure (F1)** = \(2PR/(P + R)\)
  \[
  F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}
  \]

- **Accuracy**: Fraction of docs classified correctly:
  \[
  \frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}
  \]
Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- Macroaveraging: Compute performance for each class, then average.
- Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.
### Micro- vs. Macro-Averaging: Example

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Micro Ave. Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truth: yes</td>
<td>Truth: no</td>
</tr>
<tr>
<td>Classif er: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Classif er: no</td>
<td>10</td>
<td>970</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>10</td>
<td>890</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>20</td>
<td>1860</td>
</tr>
</tbody>
</table>

- Macroaveraged precision: \((0.5 + 0.9)/2 = 0.7\)
- Microaveraged precision: \(100/120 = .83\)
Confusion Matrix

This \((i, j)\) entry means 53 of the docs actually in class \(i\) were put in class \(j\) by the classifier.

- In a perfect classification, only the diagonal has non-zero entries.
Precision-recall for category: Crude

Dumais (1998)
Precision-recall for category: Ship

Recall

Precision

LSVM
Decision Tree
Naïve Bayes
Rocchio

Dumais (1998)
Naive Bayes Classifiers

Task: Classify a new instance \( d \) based on a set of attribute values \( d = \langle x_1, x_2, \ldots, x_n \rangle \) into one of the classes \( c_j \in C \)

\[
P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}
\]

\( c_{MAP} = \arg\max_{c_j \in C} P(c_j \mid x_1, x_2, \ldots, x_n) \)

\[
= \arg\max_{c_j \in C} \frac{P(x_1, x_2, \ldots, x_n \mid c_j)P(c_j)}{P(x_1, x_2, \ldots, x_n)}
\]

MAP is “maximum a posteriori” = most likely class
Naive Bayes Classifier: Naive Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.

- $P(x_1, x_2, \ldots, x_n | c_j)$
  - $O(|X|^n \cdot |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naive Bayes Conditional Independence Assumption:

- Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i | c_j)$. 
The Naive Bayes Classifier

- **Conditional Independence Assumption:** features are independent of each other given the class:

\[ P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \cdots \cdot P(X_5 \mid C) \]
Learning the Model

simply use the frequencies in the data

\[
\hat{P}(c_j) = \frac{N(C = c_j)}{N}
\]

\[
\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}
\]
What if we have seen no training documents with the word muscle-ache and classified in the topic Flu?

\[
P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \cdots \cdot P(X_5 \mid C)
\]

What if we have seen no training documents with the word muscle-ache and classified in the topic Flu?

\[
\hat{P}(X_5 = t \mid C = Flu) = \frac{N(X_5 = t, C = Flu)}{N(C = Flu)} = 0
\]
\[ \hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k} \]

# of values of \( X_i \)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pomona</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>college</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>great</td>
<td>40</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>the</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>bad</td>
<td>2</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>is</td>
<td>98</td>
<td>99</td>
<td>98</td>
</tr>
<tr>
<td><strong>Total count</strong></td>
<td><strong>5000</strong></td>
<td><strong>5000</strong></td>
<td><strong>5000</strong></td>
</tr>
</tbody>
</table>
\[ P(\text{pos}|\text{features}) \]

\[ = P(\text{pos}) \times \text{product of probabilities } P(\text{feature}|\text{pos}) \]

\[ = P(\text{pos}) \times P(\text{"pomona"}|\text{pos}) \times P(\text{"college"}|\text{pos}) \]
\[ \quad \times P(\text{"is"}|\text{pos}) \times P(\text{"great"}|\text{pos}) \]

\[ = 0.333 \times \frac{6}{5000} \times \frac{100}{5000} \times \frac{98}{5000} \times \frac{40}{5000} \]
K Nearest Neighbor
ROCCHIO
Test Document of what class?

- Government
- Science
- Arts
Test Document = Government
Rocchio: Regions are defined by their centroid

\[ \bar{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d) \]

- Where \( D_c \) is the set of all documents that belong to class \( c \) and \( \vec{v}(d) \) is the vector space representation of \( d \).
Rocchio Properties

- Forms a simple generalization of the examples in each class (a *prototype*).
- Classification is based on similarity to class prototypes.
- It is little used outside text classification (quite effectively for text classification)
- But in general worse than Naïve Bayes
- cheap to train and test documents
Which classifier do I use for a given text classification problem?

- Is there a learning method that is optimal for all text classification problems?

Factors to take into account:

- How much training data is available?
- How simple/complex is the problem? (linear vs. nonlinear decision boundary)
- How noisy is the problem?
- How stable is the problem over time?
  - For an unstable problem, it’s better to use a simple and robust classifier.
How much data?

- Very Little:
  - There are theoretical results that Naïve Bayes should do well in such circumstances (Ng and Jordan 2002 NIPS)
  - The interesting theoretical answer is to explore semi-supervised training methods: Bootstrapping, EM over unlabeled documents, …
  - The practical answer is to get more labeled data as soon as you can

- A reasonable amount of data
  - SVMs!

- A lot of data?
  - expensive methods like SVMs (train time) or kNN (test time) are quite impractical
  - Naïve Bayes! - with lots of data, simple methods work well
Summary

- Support vector machines (SVM)
  - Choose hyperplane based on support vectors
    - Support vector = “critical” point close to decision boundary
  - (Degree-1) SVMs are linear classifiers.
  - Kernels: powerful and elegant way to define similarity metric
  - Perhaps best performing text classifier
    - But there are other methods that perform about as well as SVM, such as regularized logistic regression (Zhang & Oles 2001)
  - Partly popular due to availability of SVMlight
    - SVMlight is accurate and fast – and free (for research)
    - Now lots of good software: libsvm, TinySVM, ….
- Comparative evaluation of methods
- Real world: exploit domain specific structure!