SUPPORT VECTOR MACHINE
ACTIVE LEARNING

CS 101.2 Caltech,
03 Feb 2009

Paper by S. Tong, D. Koller
Presented by Krzysztof Chalupka
OUTLINE

- SVM intro
  - Geometric interpretation
  - Primal and dual form
  - Convexity, quadratic programming
OUTLINE

• SVM intro
  • Geometric interpretation
  • Primal and dual form
  • Convexity, quadratic programming

• Active learning in practice
  • Short review
  • The algorithms
  • Implementation
OUTLINE

- SVM intro
  - Geometric interpretation
  - Primal and dual form
  - Convexity, quadratic programming

- Active learning in practice
  - Short review
  - The algorithms
  - Implementation

- Practical results
SVM A SHORT INTRODUCTION

- Binary classification setting:
  - Input data $D_X = \{x_1, \ldots, x_n\}$, labels $\{y_1, \ldots, y_n\}$
  - Consistent hypotheses – Version Space $V$
SVM A SHORT INTRODUCTION

- SVM geometric derivation
  - For now, assume data linearly separable
  - Want to find the separating hyperplane that maximizes the distance between any training point and itself
SVM A SHORT INTRODUCTION

- SVM geometric derivation
  - For now, assume data linearly separable
  - Want to find the separating hyperplane that maximizes the distance between any training point and itself
    - Good generalization
SVM geometric derivation

- For now, assume data linearly separable
- Want to find the separating hyperplane that maximizes the distance between any training point and itself
  - Good generalization
  - Computationally attractive (later)
SVM A SHORT INTRODUCTION
SVM A SHORT INTRODUCTION

- Primal form

\[
\text{minimize}_{w,b} \quad \frac{1}{2} \|w\|^2 \\
\text{subj to } \forall_i y_i(w \cdot x_i + b) \geq 1
\]
SVM A SHORT INTRODUCTION

- Primal form

\[ \begin{align*}
\text{minimize}_{w,b} & \quad \frac{1}{2} ||w||^2 \\
\text{subj to} & \quad \forall_i y_i (w . x_i + b) \geq 1
\end{align*} \]

- Dual form (Lagrangian multipliers)

\[ \begin{align*}
\text{minimize}_{\lambda} & \quad \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{m} \lambda_i \lambda_j y_i y_j (x_i . x_j) \\
\text{subj to} & \quad \forall_i \lambda_i \geq 0 \text{ and } \sum_{i=1}^{m} \lambda_i y_i = 0
\end{align*} \]
SVM A SHORT INTRODUCTION

- Problem: classes not linearly separable
- Solution: get more dimensions
SVM A SHORT INTRODUCTION

- Get more dimensions
  - Project the inputs to a feature space

\[ f(x) = \text{sgn}(\sum_{i=1}^{m} y_i \lambda_i (\Phi(x) \cdot \Phi(x_i)) + b) \]
SVM A SHORT INTRODUCTION

- The Kernel Trick: use a (positive definite) kernel as the dot product

\[ f(x) = sgn\left(\sum_{i=1}^{m} y_i \lambda_i k(x, x_i) + b\right) \]

- OK, as the input vectors only appear in the dot product
- Again (as in Gaussian Process Optimization) some conditions on the kernel function must be met
SVM A SHORT INTRODUCTION

- Polynomial kernel
  \[ k(x, x') = (x \cdot x')^d \]

- Gaussian kernel
  \[ k(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right) \]

- Neural Net kernel (pretty cool!)
  \[ k(x, x') = \tanh(\kappa(x \cdot x') + \Theta) \]
ACTIVE LEARNING

• Recap
  • Want to query as little points as possible and find the separating hyperplane
ACTIVE LEARNING

- Recap
  - Want to query as little points as possible and find the separating hyperplane
  - Query the most uncertain points first
ACTIVE LEARNING

Recap

- Want to query as little points as possible and find the separating hyperplane
- Query the most uncertain points first
- Request labels until only one hypothesis left in the version space
ACTIVE LEARNING

Recap

- Want to query as little points as possible and find the separating hyperplane
- Query the most uncertain points first
- Request labels until only one hypothesis left in the version space
- One idea was to use a form of binary search to shrink the version space; that’s what we’ll do
ACTIVE LEARNING

- Back to SVMs
  - maximize
    \[ \text{sgn} \left( \sum_{i=1}^{m} y_i \lambda_i k(x, x') + b \right) \]

  subj to
  \[ \lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i y + i = 0 \]

- Area(V) – the surface that the version space occupies on the hypersphere \(|\mathbf{w}| = 1\) (assume \(b = 0\))
  (we use the duality between feature and version space)
ACTIVE LEARNING

- Back to SVMs
  - \( \text{Area}(V) \) - the surface that the version space occupies on the hypersphere \(|w| = 1\) (assume \(b = 0\))
    (we use the duality between feature and version space)
  - Ideally, want to always query instances that would halve \( \text{Area}(V) \)
  - \( V^+, V^- \) - the version spaces resulting from querying a particular point and getting a + or - classification
  - Want to query points with \( \text{Area}(V^+) = \text{Area}(V^-) \)
ACTIVE LEARNING

- Bad Idea
  - Compute $\text{Area}(V^-)$ and $\text{Area}(V^+)$ for each point explicitly
ACTIVE LEARNING

- Bad Idea
  - Compute \( \text{Area}(V-) \) and \( \text{Area}(V+) \) for each point explicitly

- A better one
  - Estimate the resulting areas using simpler calculations
ACTIVE LEARNING

- Bad Idea
  - Compute Area(V-) and Area(V+) for each point explicitly

- A better one
  - Estimate the resulting areas using simpler calculations

- Even better
  - Reuse values we already have
ACTIVE LEARNING

- Simple Margin
  - Each data point has a corresponding hyperplane
  - How close this hyperplane is to $w_i$ will tell us how much it bisects the current version space
  - Choose $x$ closest to $w$
ACTIVE LEARNING

○ Simple Margin
  • If $V_i$ is highly non-symmetric and/or $w_i$ is not centrally placed the result might be ugly
ACTIVE LEARNING

- MaxMin Margin
  - Use the fact that an SVMs margin is proportional to the resulting version space’s area
  - The algorithm: for each unlabeled point compute the two margins of the potential version spaces $V^+$ and $V^-$. Request the label for the point with the largest $\min(m^+, m^-)$
ACTIVE LEARNING

- MaxMin Margin
  - A better approximation of the resulting split
  - Both MaxMin and Ratio (coming next) computationally more intensive than Simple
  - But can still do slightly better, still without explicitly computing the areas
ACTIVE LEARNING

- **Ratio Margin**
  - Similar to MaxMin, but considers the fact that the shape of the version space might make the margins small even if they are a good choice.
  - Choose the point with the largest resulting:
    \[
    \min \left( \frac{m^-}{m^+}, \frac{m^+}{m^-} \right)
    \]
  - Seems to be a good choice.
ACTIVE LEARNING

● Implementation
  • Once we have computed the SVM to get $V^{+/−}$, we can use the distance of any support vector $x$ from the hyperplane

$$\left\| \sum y_i \lambda_i k(x, x_i) + b \right\|$$

to get the margins
  • Good, as many lambdas are 0s
PRACTICAL RESULTS

Article text Classification
- Reuters Data Set, around 13000 articles
- Multi-class classification of articles by topics
- Around 10000 dimensions (word vectors)
- Sample 1000 unlabelled examples, randomly choose two for a start
- Polynomial kernel classification
- Active Learning: Simple, MaxMin & Ratio
- Articles transformed to vectors of word frequencies ("bag of words")
PRACTICAL RESULTS
## PRACTICAL RESULTS

<table>
<thead>
<tr>
<th></th>
<th>Simple</th>
<th>MaxMin</th>
<th>Ratio</th>
<th>Equivalent Random size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earn</td>
<td>86.39 ± 1.65</td>
<td>87.75 ± 1.40</td>
<td>90.24 ± 2.31</td>
<td>34</td>
</tr>
<tr>
<td>Acq</td>
<td>77.04 ± 1.17</td>
<td>77.08 ± 2.00</td>
<td>80.42 ± 1.50</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Money-fx</td>
<td>93.82 ± 0.35</td>
<td>94.80 ± 0.14</td>
<td>94.83 ± 0.13</td>
<td>50</td>
</tr>
<tr>
<td>Grain</td>
<td>95.53 ± 0.09</td>
<td>95.29 ± 0.38</td>
<td>95.55 ± 1.22</td>
<td>13</td>
</tr>
<tr>
<td>Crude</td>
<td>95.26 ± 0.38</td>
<td>95.26 ± 0.15</td>
<td>95.35 ± 0.21</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Trade</td>
<td>96.31 ± 0.28</td>
<td>96.64 ± 0.10</td>
<td>96.60 ± 0.15</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Interest</td>
<td>96.15 ± 0.21</td>
<td>96.55 ± 0.09</td>
<td>96.43 ± 0.09</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Ship</td>
<td>97.75 ± 0.11</td>
<td>97.81 ± 0.09</td>
<td>97.66 ± 0.12</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Wheat</td>
<td>98.10 ± 0.24</td>
<td>98.48 ± 0.09</td>
<td>98.13 ± 0.20</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>Corn</td>
<td>98.31 ± 0.19</td>
<td>98.56 ± 0.05</td>
<td>98.30 ± 0.19</td>
<td>&gt; 100</td>
</tr>
</tbody>
</table>
PRACTICAL RESULTS
PRACTICAL RESULTS

- Usenet text classification
  - Five comp.* groups, 5000 documents, 10000 dimensions
  - 2500 randomly selected for testing, 500 of the remaining for active learning
  - Generally similar results; Simple turns out unstable
PRACTICAL RESULTS
PRACTICAL RESULTS
THE END

- SVMs for pattern classification
- Active Learning
  - Simple Margin
  - MinMax Margin
  - Ratio Margin
- All better than passive learning, but MinMax and Ratio can be computationally intensive
- Good results in text classification (also in handwriting recognition etc)