A short introduction to supervised learning, with applications to cancer pathway analysis

Dr. Christina Leslie
Computational Biology Program
Memorial Sloan-Kettering Cancer Center
http://cbio.mskcc.org/leslielab
Outline

• General ideas about supervised learning
  – (Not specific to biological domain)
  – Training, generalization, overfitting
  – Small bit of theory

• Cancer classification, gene signatures

• SVMs in some (mathematical) detail
What is machine learning?

• “Statistics with more than 20 variables”
• “Intersection of computer science and statistics”
• Provisional definition: [R. Schapire]
  – Machine learning studies how to automatically learn to make predictions based on past observations
Classification problems

- Classification:
  - Learn to classify examples into a given set of categories (“classes”)
  - Example of *supervised learning* (“labeled” training examples, i.e. known class labels)

\[
\begin{align*}
(5, "5") \quad & \quad (7, "7") \quad & \quad (2, "2") \\
\vdots 
\end{align*}
\]

new example

predicted classification

machine learning algorithm

classification rule
ML vs. “Traditional Statistics”


• “Data modeling culture” (Generative models)
  – Assume probabilistic model of known form, not too many parameters (<50)
  – Fit model to data
  – Interpret model and parameters, make predictions after
ML vs “Traditional Statistics”

• “Algorithmic modeling culture” (Predictive models)
  – Learn a prediction function from inputs to outputs, possibly many parameters (e.g. $10^2$ - $10^6$)
  – Design algorithm to find good prediction function
  – Primary goal: accurate predictions on new data, i.e. avoid overfitting, good generalization
  – Interpret after, finding “truth” is not central goal (but some “truth” in accurate prediction rule?)

• “Never solve a more difficult problem than you need to” [V. Vapnik]
Example: Generative model

![Diagram showing two probability distributions. One is labeled mean1, var1, and the other mean2, var2. The x-axis represents Voice Pitch with male and female symbols. The y-axis represents Probability. The figure is credited to Y. Freund.](image-url)
Example: Prediction function

[Figure: Y. Freund]
Poorly behaved training data

[Figure: Y. Freund]
## Conditions for accurate learning

- **Example:** predict “good” vs. “bad”  [R. Schapire]

<table>
<thead>
<tr>
<th></th>
<th>sex</th>
<th>mask</th>
<th>cape</th>
<th>tie</th>
<th>ears</th>
<th>smokes</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>batman</td>
<td>male</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>Good</td>
</tr>
<tr>
<td>robin</td>
<td>male</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Good</td>
</tr>
<tr>
<td>alfred</td>
<td>male</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>Good</td>
</tr>
<tr>
<td>penguin</td>
<td>male</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Bad</td>
</tr>
<tr>
<td>catwoman</td>
<td>female</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>Bad</td>
</tr>
<tr>
<td>joker</td>
<td>male</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Bad</td>
</tr>
<tr>
<td>batgirl</td>
<td>female</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>??</td>
</tr>
<tr>
<td>riddler</td>
<td>male</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>??</td>
</tr>
</tbody>
</table>

*Training data*

*Test data*
An example classifier

- Decision tree
Another possible classifier

- Perfectly classifies training data, makes mistakes on test set
- Intuitively too complex
Yet another classifier

- Fails to fit from training data
- Overly simple
Complexity vs. accuracy

• Classifiers must be expressive enough to capture “true” patterns in training data…

• …but if too complex, can overfit (learn noise or spurious patterns)

• Problem: Can’t tell best classifier from training error

• *Controlling overfitting* is central problem of ML
Conditions for accurate learning

• To learn an accurate classifier, need
  – Enough training examples
  – Good performance on training set
  – Control over “complexity” (Occam’s razor)

• Measure complexity by:
  – Minimum description length (number of bits needed to encode rule)
  – Number of parameters
  – VC dimension
Cancer classification

• Training data: expression data from different tumor types; few examples, high dimensional feature space

• Goals:
  – (Accurately predict tumor type)
  – Learn *gene signature* = smaller set of whose expression pattern discriminates between classes

• “Feature selection” problem
Oncogenic pathways

- [Nevins lab, Nature 2006]
- Training data:
  - Human cell cultures where specific oncogenic pathway has been activated vs. control cells (Myc, Ras, E2F3, etc)
- Prediction scores ↔ probability/confidence that pathway is activated in sample
- Test data:
  - Mouse models for pathways
  - Human cancer cell lines
Pathway signatures
Prediction in mouse models

• Rank tumors from mouse models using trained pathway vs control classifiers
Prediction scores as features

- Oncogenic pathway prediction scores used to represent tumors for clustering
- Pathway scores on cell lines correlate with response to inhibitors
Support vector machines

- SVMs are a family of algorithms for learning a *linear classification* rule from labeled training data
  \[ \{(x_1, y_1), \ldots, (x_m, y_m)\}, y_i = 1 \text{ or } -1 \]

- Well-motivated by learning theory
- Various properties of the SVM solution help *avoid overfitting*, even in very *high dimensional* feature spaces
Vector space preliminaries

- Inner product of two vectors:
  \[ <w, x> = \sum g \ w_g \ x_g \]

- Hyperplane with normal vector \( w \) and bias \( b \):
  \[ <w, x> + b = 0 \]
Linear classification rules

- SVMs consider only linear classifiers:
  \[ f_{w,b}(x) = \langle w, x \rangle + b \]
- Leads to linear prediction rules:
  \[ h_{w,b}(x) = \text{sign}(f_{w,b}(x)) \]
- Decision boundary is a hyperplane
- Prediction score \( f_{w,b}(x) \) interpreted as “confidence” in prediction
Support vector machines

- Assume linearly separable training data
- Margin of example = distance to separating hyperplane
- Margin of training set = min margin of examples
- Choose (unique) hyperplane that maximizes the margin
- Prediction score for test example \( f(x) \sim \) signed distance of \( x \) to hyperplane
• Consider training data $S$ and a particular linear classifier $f_{w,b}$

• If $\|w\| = 1$, then the geometric margin of training data for $f_{w,b}$ is

$$\gamma_S = \min_S y_i (\langle w, x_i \rangle + b)$$
Maximal margin classifier

- Hard margin SVM: given training data $S$, find linear classifier $f_{w, b}$ with \textit{maximal geometric margin} $\gamma_S$
- Solve optimization problem to find $w$ and $b$ that give maximal margin solution
Hard margin SVMs

• Equivalently, enforce a functional margin $\geq 1$ for every training vector, and minimize $||w||$

• Primal problem:
  Minimize $\frac{1}{2} <w,w>$
  subject to $y_i (<w,x_i> + b) \geq 1$
  for all training vectors $x_i$
Non-separable case

• If training data is not linearly separable, can:
  – Penalize each example by the amount it violates the margin (“soft margin SVM”)
  – Map examples to a higher dimensional space where data is separable
  – Combination of above 2 solutions
Soft margin SVMs

• Introduce slack variable $\xi_i$ to represent margin violation for training vector $x_i$

• Now constraint becomes:

$$y_i(<w, x_i>+b) \geq 1 - \xi_i$$
Soft margin SVMs

• Primal optimization problem becomes:
  Minimize
  \[
  \frac{1}{2} \langle w, w \rangle + C \sum_{i} \xi_i \quad ("1-norm") \quad \leftarrow \text{LIBSVM}
  \]
  or
  \[
  \frac{1}{2} \langle w, w \rangle + C \sum_{i} \xi_i^2 \quad ("2-norm") \quad \leftarrow \text{SVM-light}
  \]
  subject to
  \[
  y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad , \quad \xi_i \geq 0
  \]
  
• C: “trade-off” parameter
Regularization viewpoint

- Trade-off optimization problem (1-norm soft margin): minimize

\[ \|w\|^2 + C \sum_i (1 - y_i f_{w,b}(x_i))^+ \]

- \((1 - y f(x))^+\): “hinge loss”, penalty for margin violation
- \(\|w\|^2\): “regularization term”; intuitively, prevents overfitting by constraining \(w\)
Properties of SVM solution

• Introduce dual variable (“weight”) \( \alpha_i \) for each constraint, i.e. for each training example
• Solve dual optimization problem to find \( \alpha_i \)
  – Convex quadratic problem \( \rightarrow \) unique solution, good algorithms
• \( \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \)
  – Normal vector is linear combination of support vectors, i.e. training vectors with \( \alpha_i > 0 \)
Support vectors

- If $x_i$ has margin $> 1$, $\alpha_i = 0$

1-norm SVM: two kinds of support vectors
- If $x_i$ has margin $= 1$, $0 < \alpha_i < C$
- If $x_i$ has margin $< 1$, $\alpha_i = C$
Feature selection

- How to extract a “cancer signature”?
- Simplest feature selection: filter on training data
  - E.g. Apply t-test or Fisher’s criterion to find genes that discriminate between classes
  - Train SVM on reduced feature set
- Usually better to use results of training to select features
Ranking features

• Normal vector $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ gives direction in which prediction scores change
• Rank features by $|w_g|$ to get most significant components
• Recursive feature elimination (RFE): iteratively
  – Throw out bottom half of genes ranked by $|w_g|$  
  – Retrain SVM on remaining genes
Induces ranking on all genes
Kernel trick

- Idea: map to higher dimensional feature space
- Only need \textit{kernel} values: $K(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2)$ to solve dual optimization problem

```
+ + + + _
+ + + + _
+ + + + _
+ + + + _
_ _ _ _ _
```

```
_ _ _ _ _
_ _ _ _ _
_ _ _ _ _
_ _ _ _ _
```

"Input Space" \hspace{1cm} \Phi \hspace{1cm} "Feature Space"
Examples of kernels

- Large margin non-linear decision boundaries
- Not needed with expression data

Degree 2 polynomial

\[ K(x,z) = (x \cdot z + C)^2 \]

Radial basis

\[ K(x,z) = \exp \left( -\frac{\|x - z\|^2}{\sigma^2} \right) \]
Discussion issues for paper

• How well-defined is a cancer signature?
  – How stable is feature selection on small data set?
  – Empirical validation of gene set, number of genes?

• Which analyses are purely training data results, which show prediction performance?

• Significance of prediction performance?
  – Traditional ML does not assert significance via a $p$-value but comparison against other methods
  – Can compare to a baseline method, e.g. single oncogene expression level
Be careful!

• Potti et al., Nature Medicine 2006: Similar analysis to predict response to chemotherapy, based on NCI 60 cell line data

• Coombed et al., Nature Medicine 2007: “Bioinformatics forensics”, unable to reproduce results
  – Mislabeling of samples (+ vs -)
  – Off-by-one indexing error, wrong genes in signature
  – No separation of training and test for feature reduction (“metagene”), not strictly inductive learning

• Summary: poor computational practices and (probably) overfitting lead to erroneous results