Review

Problem: Given a set of linearly separable points find the weights of a homogenous LTF that separates points.

- **Perceptron Learning Algorithm**
  - Capabilities and convergence
  - Ill posed problem (version space, infinite number of solutions)
  - Extensions (nonzero thresholds, transformation of inputs)

- **Optimum Margin Classifiers**
  - Maximize margins, solving QP problems, cost function (hinge loss, quadratic loss)
  - Well posed problem
Alternative Characterization of Optimal Margin Classifiers

Maximizing margins equivalent to minimizing magnitude of weight vector.

\[ \mathbf{w}^T (\mathbf{u} - \mathbf{v}) = 2 \]

\[ \mathbf{w}^T (\mathbf{u} - \mathbf{v}) / \| \mathbf{w} \| = 2 / \| \mathbf{w} \| = 2m \]

\[ \mathbf{w}^T \mathbf{u} + b = 1 \]

\[ \mathbf{w}^T \mathbf{v} + b = -1 \]
Quadratic Programming Problem

Problem statement:
\[ \max_{(w,b)} \min \left( \| x - x(i) \| \right) : x \in \mathbb{R}^n, \ w^T x + b = 0, \ 1 \leq i \leq l \]

which is equivalent to solving (QP) problem:
\[ \text{Min } \frac{1}{2} \| w \|^2 \]
subject to \( d(i) (w^T x(i) + b) \geq 1, \ \forall i \)
Lagrange Multipliers

Constrained optimization can be dealt with by introducing Lagrange multipliers $\alpha(i) \geq 0$ and a Lagrangian

$$L(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum \alpha(i) \left(d(i) (w^T x(i) + b) - 1\right)$$
Support Vectors

QP program is convex and note that at solution \( \partial L(w,b,\alpha) / \partial w = 0 \) and \( \partial L(w,b,\alpha) / \partial b = 0 \) leading to

\[ \sum \alpha(i) d(i) = 0 \quad \text{and} \quad w = \sum \alpha(i) d(i) x(i) \]

Weight vector expressed in terms of subset of training vectors \( x_i \) and \( \alpha_i \) that are nonzero. These are support vectors. By KKT conditions we have that

\[ \alpha(i) (d(i) (w^T x(i) + b) - 1) = 0 \]

indicating that support vectors lie on margin.
Points in red are support vectors.
Dual optimization representation

Solving the primal QP problem is equivalent to solving the dual QP problem. Primal variables $w, b$ are eliminated.

$$\max \ W(\alpha) = \Sigma \alpha(i) - \frac{1}{2} \Sigma \alpha(i) \alpha(j) d(i) d(j)(x(i)^T x(j))$$
subject to $\alpha(i) \geq 0$, and $\Sigma \alpha(i) d(i) = 0$.

The hyperplane decision function can be written as

$$f(x) = \text{sgn} \ (\Sigma \alpha(i) d(i) x^T x(i) + b)$$
Comments about SVM solution

- Solution can be found in primal or dual spaces. When we discuss kernels later there are advantages of finding solution in dual space.
- Threshold, b is determined after weight w is found.
- In version space, SVM solution is center of largest hypersphere that fits in version space.
- Solution can be modified to
  - Handle training data that is not linearly separable via slack variables
  - Implement nonlinear discriminant function via kernels
- Solving quadratic programming problems
Handling data that not linearly separable

- Data is often noisy or inconsistent resulting in training data being not linearly separable.
- PLA algorithm can be modified (pocket algorithm), however there are problems with algorithm termination.
- SVM can easily handle this case by adding slack variables to handle pattern recognition.
Gaussian data drawn from two classes
Training examples below margin

margins

Optimal hyperplane
Formulation with slack variables

Optimal margin classifier with slack variables and kernel functions described by Support Vector Machine (SVM).

\[
\begin{align*}
\min_{(w, \xi)} & \quad \frac{1}{2}\|w\|^2 + \gamma \sum \xi(i) \\
\text{subject to} & \quad \xi(i) \geq 0 \quad \forall i, \quad d(i)(w^T x(i) + b) \geq 1 - \xi(i), \\
& \quad \forall i, \text{ and } \gamma > 0.
\end{align*}
\]

In dual space

\[
\begin{align*}
\max W(\alpha) = & \quad \sum \alpha(i) - \frac{1}{2} \sum \alpha(i)\alpha(j) d(i) d(j) x(i)^T x(j) \\
\text{subject to} & \quad \gamma \geq \alpha(i) \geq 0, \text{ and } \sum \alpha(i) d(i) = 0.
\end{align*}
\]

Weights can be found by \( w = \sum \alpha(i) d(i) x(i) \).
Solving QP Problem

- Quadratic programming problem with linear inequality constraints.
- Optimization problem involves searching space of feasible solutions (points where inequality constraints satisfied).
- Can solve problem in primal or dual space.
QP software for SVM

- Matlab (easy to use, choose primal or dual space, slow): quadprog()
  - Primal space (w, b, \( \gamma^+, \gamma^- \))
  - Dual space (\( \alpha \))

- Sequential Minimization Optimization (SMO) (specialized for solving SVM, fast): decomposition method, chunking method

- SVM light (fast): decomposition method
Example

Drawn from Gaussian data $\text{cov}(X) = I$

$20 + \text{pts. Mean} = (.5,.5)$

$20 - \text{pts. Mean} = -(0.5,0.5)$
Example continued

Primal Space (matlab)

```matlab
x = randn(40,2);
d = [ones(20,1); -ones(20,1)];
x = x + d * [.5 .5];
H = diag([0 1 1 zeros(1,80)]);
gamma = 1;
f = [zeros(43,1); gamma*ones(40,1)];
Aeq = [d x.*(d*[1 1]) -eye(40) eye(40)];
beq = ones(40,1);
A = zeros(1,83);
b = 0;
lb = [-inf*ones(3,1); zeros(80,1)];
ub = [inf*ones(83,1)];
[w,fval] = quadprog(gamma*H,f,A,b,Aeq,beq,lb,ub);
```

Example continued

Dual Space (matlab)

\[ \text{xn} = x .* (d*[1 1]); \]
\[ k = \text{xn} \times \text{xn}'; \]
\[ \text{gamma} = 1; \]
\[ f = -\text{ones}(40,1); \]
\[ \text{Aeq} = d'; \]
\[ \text{beq} = 0 \]
\[ \text{A} = \text{zeros}(1,40); \]
\[ \text{b} = 0; \]
\[ \text{lb} = [\text{zeros}(40,1)]; \]
\[ \text{ub} = [\text{gamma} \times \text{ones}(40,1)]; \]
\[ [\text{alpha}, \text{fvala}] = \text{quadprog}(k, f, A, b, \text{Aeq}, \text{beq}, \text{lb}, \text{ub}); \]
Example continued

- \( w = (1.4245, .4390)^T \ b = 0.1347 \)
- \( w = \sum \alpha(i) \ d(i) \ x(i) \) (26 support vectors, 3 lie on margin hyperplane)
  - \( \alpha(i) = 0, \ x(i) \) above margin
  - \( 0 \leq \alpha(i) \leq \gamma, \ x(i) \) lie on margin hyperplanes
  - \( \alpha(i) = \gamma, \ x(i) \) lie below margin hyperplanes
- Hyperplane can be represented in
  - Primal space: \( w^T x + b = 0 \)
  - Dual space: \( \sum \alpha(i) \ d(i) \ x^T x(i) + b = 0 \)
- Regularization parameter \( \gamma \) controls balance between margin and errors.
Sequential Minimization Optimization breaks up QP program into small subproblems that are solved analytically.

- SMO solves dual QP SVM problem by examining points that violate KKT conditions.
- Algorithm converges and consists of:
  - Search for 2 points that violate KKT conditions.
  - Solve QP program for 2 points.
  - Calculate threshold value b.
- Continue until all points satisfy KKT conditions.
- Convergence time depends on difficulty of classification problem and kernel functions used.
SVM Light

- Decomposition method: partition data into small working set and fixed set. Only learn Lagrangian multipliers associated with training data belonging to working set.
- Software available at Thorsten Joachim’s website at Cornell University.
- Data represented to take advantage of possible sparseness, nonnumeric training data.
References

- Kernel website: http://www.kernel-machines.org/