Topics

- Regularization cont’d
  - regularized logistic regression
  - empirical vs. expected loss

- Support vector machine (part 1)
  - discrimination, “optimal” hyperplane
  - optimization via Lagrange multipliers
Review: “choices” in logistic regression

- **Simple logistic regression model**

  \[ P(y = 1|x, w) = g(w_0 + w_1 x) \]

  parameterized by \( w = (w_0, w_1) \). We assume that \( x \in [-1, 1] \), i.e., that the inputs remain bounded.

- We can now divide the parameter space into regions with centers \( w_1, w_2, \ldots \) such that the predictions of any \( w \) (for any \( x \in [-1, 1] \)) are close to those of one of the centers:

  \[ | \log P(y = 1|x, w) - \log P(y = 1|x, w_j) | \leq \epsilon \]
Review: regularized logistic regression

• We can regularize by imposing a penalty in the estimation criterion that encourages $\|w\|$ to remain small.

Maximum penalized likelihood

$$l(D; w, \lambda) = \sum_{i=1}^{n} \log P(y_i|x_i, w) - \frac{\lambda}{2} \|w\|^2$$

where larger values of $\lambda$ impose stronger regularization.

• More generally, we can assign penalties based on prior distributions over the parameters, i.e., add $\log P(w)$ in the log-likelihood criterion.
Effect of available “choices”

- We’d like the empirical loss of our parameter estimate $\hat{w}$ to be close to its expected loss

Example: $m$ effective parameter choices

$$L_n(w_k) = \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, f(x_i, w_k)), \quad k = 1, \ldots, m$$

$$L_n(\hat{w}) = \min_{i} \{ L_n(w_i) \}$$
Empirical vs expected loss

- How is $\min_i \{ L_n(w_i) \}$ distributed in the simple case where each

$$L_n(w_k) = \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, f(x_i, w_k)),$$

is a zero mean Gaussian?
Topics

- Support vector machine
  - discrimination, “optimal” hyperplane
  - optimization via Lagrange multipliers
  - kernel function
Discriminative (non-probabilistic) classification

- Consider a binary classification task with $y = \pm 1$ labels (not 0/1 as before). When the training examples are \textit{linearly separable} we can set the parameters of a linear classifier so that all the training examples are classified correctly:

$$y_i \left[ w_0 + w^T x \right] > 0, \quad i = 1, \ldots, n$$

The label we predict for each example is given by the sign of the linear function $w_0 + w^T x$. 

Tommi Jaakkola, MIT AI Lab
Classification and margin

- We can try to find a unique solution by requiring that the training examples are classified correctly with a non-zero “margin”

\[ y_i \left[ w_0 + w^T x_i \right] - 1 \geq 0, \ i = 1, \ldots, n \]

The margin should be defined in terms of the distance from the boundary to the examples rather than based on the value of the linear function.
Redefining margin

- One dimensional example: \( f(x; w_1, w_0) = w_0 + w_1 x \).

Relevant constraints:

\[
1 \left[ w_0 + w_1 x^+ \right] - 1 \geq 0 \\
-1 \left[ w_0 + w_1 x^- \right] - 1 \geq 0
\]
Redefining margin

- One dimensional example:  \( f(x; w_1, w_0) = w_0 + w_1 x \).

Relevant constraints:

\[
1 [w_0 + w_1 x^+] - 1 \geq 0 \\
-1 [w_0 + w_1 x^-] - 1 \geq 0
\]

By adding the two inequalities we get

\[
w_1 (x^+ - x^-) - 2 \geq 0 \\
\frac{|x^- - x^+|}{2} \geq \frac{1}{|w_1|}
\]

We get maximum margin separation by minimizing \(|w_1|\)
Support vector machine

- We minimize a regularization penalty

\[ \|w\|^2/2 = w^T w / 2 = \sum_{j=1}^{d} w_j^2 / 2 \]

subject to the classification constraints

\[ y_i [w_0 + w^T x_i] - 1 \geq 0, \quad i = 1, \ldots, n \]

- The attained margin is now given by \( 1/\|w\| \)
- Only a few of the classification constraints are relevant
  \( \Rightarrow \) support vectors
Support vector machine cont’d

- We find the optimal setting of \( \{w_0, w\} \) by introducing Lagrange multipliers \( \alpha_i \geq 0 \) for the inequality constraints.

- We minimize

\[
J(w, w_0, \alpha) = \|w\|^2 / 2 - \sum_{i=1}^{n} \alpha_i \left( y_i [w_0 + w^T x_i] - 1 \right)
\]

with respect to \( w, w_0 \). \( \{\alpha_i\} \) ensure that the classification constraints are indeed satisfied.

For fixed \( \{\alpha_i\} \)

\[
\frac{\partial}{\partial \mathbf{w}} J(w, w_0, \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0
\]

\[
\frac{\partial}{\partial w_0} J(w, w_0, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0
\]
Solution

• Substituting the solution \( w = \sum_{i=1}^{n} \alpha_i y_i x_i \) back into the objective leaves us with the following (dual) optimization problem over the Lagrange multipliers:

We maximize

\[
J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)
\]

subject to the constraints

\[
\alpha_i \geq 0, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} \alpha_i y_i = 0
\]

(For non-separable problems we have to limit \( \alpha_i \leq C \))

• This is a quadratic programming problem
Support vector machines

- Once we have the Lagrange multipliers \( \{\hat{\alpha}_i\} \), we can reconstruct the parameter vector \( \hat{\mathbf{w}} \) as a weighted combination of the training examples:

\[
\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i
\]

where the “weight” \( \hat{\alpha}_i = 0 \) for all but the support vectors (SV).

- The decision boundary has an interpretable form

\[
\hat{\mathbf{w}}^T \mathbf{x} + \hat{w}_0 = \sum_{i \in SV} \hat{\alpha}_i y_i (\mathbf{x}_i^T \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)
\]
Interpretation of support vector machines

- To use support vector machines we have to specify only the inner products (or *kernel*) between the examples $(\mathbf{x}_i^T \mathbf{x})$
- The weights $\{\alpha_i\}$ associated with the training examples are solved by enforcing the classification constraints.  
  $\Rightarrow$ sparse solution

- We make decisions by comparing each new example $\mathbf{x}$ with **only** the support vectors $\{\mathbf{x}_i\}_{i \in SV}$:

$$
\hat{y} = \text{sign} \left( \sum_{i \in SV} \hat{\alpha}_i y_i (\mathbf{x}_i^T \mathbf{x}) + \hat{w}_0 \right)
$$