Statistical Techniques in Robotics (16-831, S22) Lecture #09 (Wednesday, Feb 16)

Online Gradient Descent

Lecturer: Kris Kitani *Scribes: Siqi Chai, Jiajie Xu (Group D)*

1 Review

In the previous lecture, we discussed the relationship between Follow The Regularized Leader (FTRL) and Online Mirror Descent (OMD) algorithms. We showed that they are equivalent through reparametrizing dual parameters and the mirror function. We introduced the concept of duality and derived the regret bound for OMD. In this lecture, we will cover Gradient Descent and Stochastic Gradient Descent. In the online scenario, gradient descent is a type of OMD, and we will derive its regret bound.

1.1 Mirror Function

We recall that in order to show the equivalence between FTRL and OMD, we introduce the dual space. In FTRL, we optimize weight parameters $w*$ in the primal space. In OMD, We optimize in the dual space and then map to primal space. Since the dual space is induced by the regularization term, $\psi(w)$, the way we map from dual space back to primal space depends on the way we regularize. The mirror function, $g(\theta)$ characterizes the mapping. In FTRL when we have a linear loss and convex regularization, we predict w as follow:

$$w^{(t+1)} = arg\min_{w} < w, z^{(1:t)} > +\psi(w)$$

we can convert it to a maximization problem by mapping to the dual space:

$$\theta^{(t+1)} := -z^{(1:t)}$$

The optimization problem now becomes:

$$\begin{aligned} w^{(t+1)} &= arg\max_{w} < w, \theta^{(t+1)} > -\psi(w) \\ &= g(\theta^{(t+1)}) \end{aligned}$$

This is how we convert a FTRL with linear loss and convex regularization into a OMD problem!

1.2 OMD Regret Bound

Definition 1. Convex Conjugate:

$$\psi^*(\theta) = max_w(< \theta, w >, -\psi(w))$$

The terms to be maximized is geometrically the difference between a function's value at w and the line with slope of the function at w. At the optimal w, the conjugate functional evaluates to the intercept of tangent of $\psi(w)$.

Fenchel-Young Inequality:

$$\psi(\theta) \geq (< w, \theta > -\psi(w))$$

The proof for this comes directly from the definition of Convex Conjugate. When w takes whatever non-optimal value, the left hand side will evaluate to be grater than the right hand side.

Bregman Divergence:

$$D_\psi(w||u) = \psi(w) - \psi(u) - \nabla\psi(u)^T(w - u)$$

The Bregman Divergence equation can be interpreted as a measurement of approximation error. The first two terms evaluates to the difference between function values at two distinct points. The last term is the first order approximation of the function. Note that ψ is a convex function.

OMD regret bound:

$$R(u) = \sum_{t=1}^{T} w^{(t)} \cdot z^{(t)} - u \cdot z^{(t)}$$

Armed with the above tools, we're now ready to derive the regret bound for OMD defined above. By substituting in the dual parameter and applying Fenchel-Young Inequality we get:

$$\psi(u) - u \cdot \theta^{(T+1)} \geq -\psi^*(\theta^{(T+1)})$$

We show by telescoping technique that:

$$\psi(\theta^{(1)}) = -\psi(w^{(1)})$$

Finally by some algebra and Bregman Divergence, we have the upper bound of reget as:

$$R(u) \leq \psi(u) - \psi(w^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(-z^{1:t}|| - z^{1:t-1})$$

1.3 Main Take Away

- FTRL can be interpreted as OMD.
- The Mirror function depends on the regularization function, which then affects the regret bound.
- OMD is a generic algorithm to solve for online convex optimization.

2 Summary

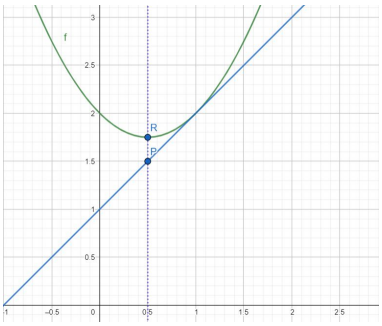
2.1 Gradient Descent: 3 interpretations

Algorithm 1 Gradient Descent

1: $\mathbf{w}^{(0)} \leftarrow 0$
2: **for** $t = 1, \cdots, T$ **do**
3: COMPUTE $(\nabla f(w^{(t-1)}))$
4: $w^{(t)} = w^{(t-1)} - \eta \nabla f(w^{(t-1)})$ ▷ Weight update
5: **end for**

We are familiar with the Gradient Descent algorithm, but we may not know the reason behind it. It is intuitive to update the weights as in the pseudo-code: walking down the gradient, but we need some mathematical interpretations. There are three:

Interpretation 1: Geometric



From the geometric perspective, the optimal parameter always stay at the bottom of the 'hill', and that's why we should walk to the opposite direction of the gradient. Note that the gradient, which is the derivative of the objective function with respect to weight parameters, can be a scalar, a vector, or even a matrix. It depends on what kind of objective function we have, and also the form of weight parameter.

$$\nabla f(w) = \{\frac{\delta f(w)}{\delta w_1}, \cdots, \frac{\delta f(w)}{\delta w_N}\}$$

Interpretation 2: Linear Approximation

The geometric interpretation is a natural but not rigorous. Here, we give another more mathematical interpretation. We formulate the problem as a linear approximation problem. That is, we approximate the optimal weight, which gives the minimum loss, by first order Taylor approximation.

$$f(u) \approx f(w) + < u - w, \nabla f(w) >$$

In the convex setting, we formulate the lower bound of $f(u)$ as:

$$f(u) \geq f(w) + < u - w, \nabla f(w) >$$

We want to find the optimal w, that minimize the loss, thus we take the above inequality and minimize the right hand side. When we do so, however, we find that the result is negative infinity! Recall that the Taylor approximation only works for a small step size, thus we need to constraint w so that it is close to the previous value before update.

$$\min_w ||w - w^{(t)}||_2^2$$

With this constraint applied, the optimization function now have two parts:

- minimize with linear approximation
- minimize the step size

Thus the final optimization formulation is:

$$w^{(t+1)} = argmin_w \eta \frac{1}{2}||w - w^{(t)}||^2 + (f(w^{(t)}) + < w - w^{(t)}, \nabla f(w^{(t)}) >)$$

Note that now we have a linear loss and a quadratic regularization!

Interpretation 3: Isometric Quadratic Approximation

In the previous section, we applied the first order Taylor approximation. We can take it further by using the second order Taylor approximation. That is, we approximate the gradient with second order derivative, and reduce the approximation error. The new optimization looks like:

$$f(u) \approx f(w) + (u - w)^T \nabla f(w) + \frac{1}{2}(u - w)^T \nabla^2 f(w)(u - w)$$

Again, in a convex condition, the right hand side is the lower bound of approximation:

$$f(u) \geq f(w) + (u - w)^T \nabla f(w) + \frac{1}{2}(u - w)^T \nabla^2 f(w)(u - w)$$

Generally speaking, computing the second order derivative with respect to w is expensive. Assume w is a n-d vector:

$$O(hessian) = O(n^2)$$
$$O(derivative) = O(n)$$

This computation takes quadratic time and we need to reduce it! Here we simplify the hessian matrix to a identity matrix:

$$f(u) \geq f(w) + (u - w)^T \nabla f(w) + \frac{1}{2\eta}(u - w)^T I(u - w)$$

We will be using this formulation later in Support Vector Machine (SVM).

Solving the optimization

We proceed to solve the optimization problem defined in prospective 2.

$$w^{(t+1)} = argmin_w \eta \frac{1}{2}||w - w^{(t)}||^2 + (f(w^{(t)}) + < w - w^{(t)}, \nabla f(w^{(t)}) >)$$

Our goal is to find a better weight, $w^{(t+1)}$, that will reduce the loss function. In order to solve it, we take the derivative and set the derivative to 0 (refer to Appendix 3.1 for details):

$$w = w^{(t)} - \eta \nabla f(w^{(t)})$$

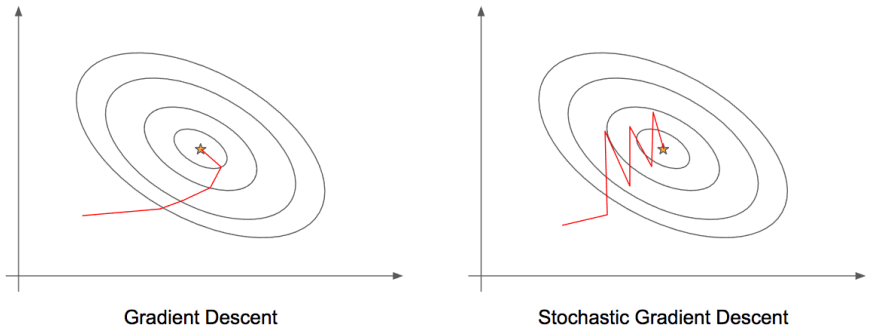
Now we have it! The above weight update function was seen in the Weighted Majority Algorithm (WMA), the Online Perceptron Algorithm, as well as the Follow The Regularized Leader algorithm (FTRL).

2.2 Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descent

1: $\mathbf{w}^{(1)} \leftarrow 0$
2: $\eta > 0$
3: **for** $t = 1, \cdots, T$ **do**
4: SAMPLE $(z \sim D)$ ▷ Sample from data distribution
5: COMPUTE $(\nabla f_z(w^{(t-1)}))$ ▷ Fast to compute
6: $w^{(t)} = w^{(t-1)} - \eta \nabla f_z(w^{(t-1)})$ ▷ Weight update
7: **end for**

Each iteration of gradient descent update can be computationally expensive given a large dataset. In fact, as long as we can eventually arrive at the bottom of the 'hill', we do not need to always take the globally optimal path. That is, we can take gradient descent steps according to a small batch of data as long as the expected direction equal the gradient direction. We will save a lot computation if we only calculate the gradient on a mini-batch. Still, the convergence bound of SGD is smiliar to that of gradient descent.



Theorem 2. *(SGD/OGD is special case OMD)*

Algorithm 3 Online Sub-Gradient Descent

1: **for** $t = 1, \cdots, T$ **do**
2: $\theta^{(t+1)} = \theta^{(t)} - z^{(t)}, z^{(t)} \in \delta f^{(t)}(w^{(t)})$ ▷ Dual Parameter update with magnitude assumption
3: $w^{(t+1)} = \eta\theta^{(t+1)}$ ▷ Mirror projection
4: **end for**

Algorithm 4 Online Proj Sub-Gradient Descent

1: **for** $t = 1, \cdots, T$ **do**
2: $\theta^{(t+1)} = \theta^{(t)} - z^{(t)}, z^{(t)} \in \delta f^{(t)}(w^{(t)})$ ▷ Dual Parameter update
3: $w^{(t+1)} = \Pi_{(\theta \to s)}\eta\theta^{(t+1)}$ ▷ Mirror projection with explicit feasibility condition
4: **end for**

Proof. Define a OMD with linear loss can quadratic regularization, show the mirror function is same as OGD:
Quadratic Regularization:

$$\psi(w) = \frac{1}{2\eta}||w||_2^2$$

Linear Loss:

$$f(w) =< w, \theta >$$

Then we have a prediction rule:

$$\begin{aligned} w^{(1+t)} &= arg\min_w < w, -\theta^{(1+t)} > + \frac{1}{2\eta}||w||_2^2 \\ &= arg\min_w < w, -\theta^{(1+t)} > + \frac{1}{2\eta}\sum_n w_n^2 \end{aligned} \tag{1}$$

We define the right hand side as a loss function, and minimize it by taking the derivative:

$$L =< w, -\theta^{(1+t)} > + \frac{1}{2\eta}\sum_n w_n^2$$

$$\frac{\delta L}{\delta w_n} = -\theta_n + \frac{1}{2\eta}2w_n = 0$$

$$w_n = \eta\theta_n$$

Recall that the OMD mirror function is:

$$g(\theta) = \eta\theta$$

□

These are exactly the same! Thus, OGD is a special case of OMD. Note that we gave two versions of the algorithm: Online Sub-Gradient Descent (Alg.3) and Online Proj Sub-Gradient Descent. The difference between the two is that in Online Sub-Gradient Descent we do not need to actually project from dual space to primal space. That is, on line 3 of the pseudo code, we update w by multiplied θ. The simplicity comes from the assumption that the dual space gradient has a constrained magnitude. In the Online Proj Sub-Gradient Descent, we need to project from dual space to primal space with the funtion $\Pi_{(\theta \to s)}$. For example, we can use the eucilidian projection:

$$\Pi_{(\theta \to s)}(x) = arg\min_y ||x - y||^2$$

2.3 OGD Regret Bound

Theorem 3. *(Regret Bound of Online Gradient Descent) with assumed maximum magnitude of primal parameter D and assumed maximum magnitude of dual parameter sub-gradient G*

$$R_{OGD} \leq DG\sqrt{T}$$

where

$$\psi(w) = \frac{1}{2\eta}||w||_2^2$$

$$D = max||u||^2, u \in S$$

$$G = max||z||^2, u \in \delta f(w)$$

Lemma 4. *Under L2 norm, the convex conjugate function is in the same form as the regularizing function:*

$$\psi(w) = \frac{1}{2}||w||_2^2$$

$$\psi^*(\theta) = \frac{1}{2}||\theta||_2^2$$

Proof.

$$\begin{aligned} \psi^*(\theta) &= \max_w(< \theta, w > -\psi(w)) \\ &= -\min_w(\psi(w) - < \theta, w >) \end{aligned} \tag{2}$$

$$\begin{aligned} -\psi^*(\theta) &= \min_w(\psi(w) - < \theta, w >) \\ &= \min_w(\frac{1}{2}||w||_2^2 - < \theta, w >) \end{aligned} \tag{3}$$

To minimize, we take the derivative and set it to 0

$$\begin{aligned} \frac{\delta}{\delta w}\{\frac{1}{2}||w||_2^2 - < \theta, w >\} &= 0 \\ w - \theta &= 0 \\ w &= \theta \end{aligned} \tag{4}$$

Now we plug in w to the convex conjugate function (refer to Appendix 3.2):

$$\psi^*(\theta) = \frac{1}{2}||\theta||_2^2$$

□

Proof. (Regret bound of OGD) As shown previously, OGD is a special case of OMD, thus we start by the regret formulation for OMD:

$$R(u) \leq \psi(u) - \psi(w^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(\theta^{(t+1)}||\theta^{(t)})$$

By Bregman Divergence, we can substitute the last term:

$$R(u) \leq \psi(u) - \psi(w^{(1)}) + \sum_{t=1}^{T} \psi^*(\theta^{(t+1)}) - \psi^*(\theta^{(t)}) - \nabla\psi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})$$

We plug in the defined regularizing function and derived convex conjugate function from Lemma 4 (Algebra in Appendix 3.3):

$$R(u) \leq \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}||\theta^{(t+1)}||_2^2 - \frac{1}{2\eta}||\theta^{(t)}||_2^2 - \nabla\frac{1}{2\eta}||\theta^{(t)}||_2^2(\theta^{(t+1)} - \theta^{(t)}) \tag{5}$$

By update rule of θ:

$$\theta^{(t+1)} = \theta^{(t)} - \eta z^{(t)}$$

We have (Algebra in Appendix 3.4):

$$R(u) \leq \frac{1}{2\eta}||u||_2^2 + \sum_{t=1}^{T} \frac{\eta}{2}||z^{(t)}||_2^2 \tag{6}$$

Recall the definition of D and G:

$$D = max||u||^2, u \in S$$
$$G = max||z||^2, u \in \delta f(w)$$

Plugging in we have:

$$\begin{aligned} R(u) &\leq \frac{1}{2\eta}||u||_2^2 + \sum_{t=1}^{T} \frac{\eta}{2}||z^{(t)}||_2^2 \\ &\leq \frac{D^2}{2\eta} + \frac{\eta}{2}TG^2 \end{aligned} \tag{7}$$

We need to figure out the optimal step size η. Again, we take the derivative and set it to 0 (Appendix 3.5):

$$\frac{\delta}{\delta\eta}\frac{D^2}{2\eta} + \frac{\eta}{2}TG^2 = 0$$
$$\eta = \frac{D}{G\sqrt{T}}$$

Now we plug the optimal η back and solve for the regret bound:

$$R(u) = DG\sqrt{T} \tag{8}$$

□

We've completed the proof of OGD regret bound.

2.4 Online Normalized Exponentiated Gradient Descent

Online normalized exponentiated gradient descent (ONEGD) is an specific type of online mirror descent, with linear loss and entropic regularization. In practice, it may give the advantage of normalzing gradient make it more stable. Algorithm pseudocode:

Algorithm 5 Online Norm-Exponentiated-Gradient(η)

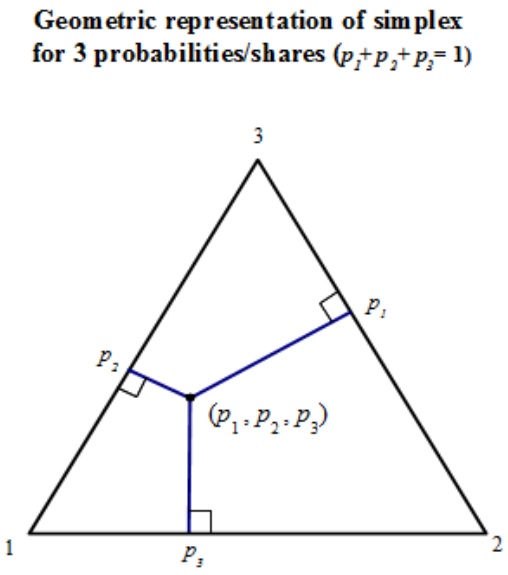
1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
3: $\quad \boldsymbol{w}^{(t+1)} \propto \boldsymbol{w}^{(t)} \exp\left(\eta \boldsymbol{z}^{(t)}\right)$ ▷ Mirror projection
4: **end for**

2.4.1 Loss and regularization setup

Define regularization term as negative entropy with K-simplex constraint:

$$\psi(\boldsymbol{w}) = \sum_{k=1}^{K} w_k \log w_k \quad \boldsymbol{w} \in \mathbb{S}^K$$

This constraint is used when solution space is a probability simplex. The simplex constraint means each dimension of $\boldsymbol{w}$ sums up to 1: $\sum_k w_k = 1$. In the example below where $K = 3$, solution space of $\boldsymbol{w}$ is inside the triangle.



[4]

Define loss function as linear:

$$f(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle$$

Recall, the parameter update equation for online mirror descent, substitute regularization term:

$$\begin{aligned}\boldsymbol{w}^{(t+1)} &= \arg\min_{\boldsymbol{w}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \psi(\boldsymbol{w}) \\ &= \arg\min_{\boldsymbol{w} \in \mathbb{S}^K} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} w_k \log w_k\end{aligned}$$

Add simplex constraint to the parameter update as Lagrange multiplier:

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w} \in \mathbb{S}^K} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} w_k \log w_k + \lambda \left(1 - \sum_k w_k\right)$$

When adding simplex constraint to objective function, we can rewrite it as Lagrangian expression:

$$\mathcal{L} = \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \frac{1}{\eta} \sum_{k=1}^{K} w_k \log w_k + \lambda \left(1 - \sum_k w_k\right)$$

$\frac{1}{\eta}$ here is to balance the weight between loss and regularization term.

2.4.2 Minima of Lagrangian

To solve the minima of $\mathcal{L}$, take its gradient to $\boldsymbol{w}$ and set it equal to 0. First, calculate partial derivative with respect to each element of $\boldsymbol{w}$:

$$\frac{\partial \mathcal{L}}{\partial w_n} = -\theta_n + \frac{1}{\eta}(1 + \log w_n) - \lambda = 0$$

Solve for w_n:

$$w_n = \frac{\exp(\eta \theta_n)}{\exp(1 - \eta\lambda)}$$

Dual parameter update of OMD:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$$

Recall, definition of mirror/linking function: $\boldsymbol{w} = g(\boldsymbol{\theta})$, combine the partial derivatives. Mirror function enforces geometry of problem, we have:

$$g(\boldsymbol{\theta}) = \frac{\exp(\eta \boldsymbol{\theta})}{\sum_k \exp(\eta \theta_k)}$$

To get the update equation for primal parameter $\boldsymbol{w}$ in incremental form, we start from this mirror function. The key here is to substitute incremental form of dual parameter $\boldsymbol{\theta}$, and multiply a

temporary term with the same sum of exponential on numerator and denominator:

$$
\begin{aligned}
w_n^{(t+1)} =& \frac{\exp(\eta\theta_n^{(t+1)})}{\sum_k \exp(\eta\theta_k)} \\
=& \frac{\exp\left(\eta(\theta_n^{(t)} - z_n^{(t)})\right)}{\sum_k \exp\left(\eta(\theta_k^{(t)} - z_k^{(t)})\right)} && \triangleright \text{incremental form of } \theta \\
=& \exp(\eta\theta_n^{(t)}) \frac{\exp(-\eta z_n^{(t)})}{\sum_k \exp(\eta\theta_k^{(t)}) \exp(-\eta z_k^{(t)})} \cdot \frac{\sum_j \exp(\eta\theta_j^{(t)})}{\sum_j \exp(\eta\theta_j^{(t)})} && \triangleright \text{temporary term} \\
=& \frac{\exp(\eta\theta_n^{(t)})}{\sum_j \exp(\eta\theta_j^{(t)})} \cdot \exp(-\eta z_n^{(t)}) \cdot \frac{\sum_j \exp(\eta\theta_j^{(t)})}{\sum_k \exp(\eta\theta_k^{(t)}) \exp(-\eta z_k^{(t)})} && \triangleright \text{rearrange terms} \\
=& \frac{w_n^{(t)} \exp(-\eta z_n^{(t)})}{\sum_k w_k^{(t)} \exp(-\eta z_k^{(t)})} && \triangleright \text{substitute with def of } w
\end{aligned}
$$

Therefore, we can get: $w_n^{(t+1)} \propto w_n^{(t)} \exp(-\eta z_n^{(t)})$
This is the same exponential update rule as weighted majority algorithm.

2.4.3 Connection to Hedge algorithm

The hedge algorithm is similar to the weighted majority algorithm, with different exponential update rule. Hedge is an unnormalized exponentiated gradient descent algorithm.

Algorithm 6 Hedge Algorithm(β)

1: $\boldsymbol{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^N$
2: **for** $t = 1, \cdots, T$ **do**
3: RECEIVE ($\boldsymbol{x}^{(t)} \in \{-1, 1\}$)
4: $i \sim$ MULTINOMIAL($\boldsymbol{w}^{(t)}/\Phi^{(t)}$), $\Phi^{(t)} = \sum_{n=1}^N w_n^{(t)}$
5: $\hat{y}^{(t)} = h_i(\boldsymbol{x}^{(t)})$ ▷ Expect 0-1 loss, linear
6: RECEIVE ($y^{(t)} \in \{-1, 1\}$)
7: $w_n^{(t+1)} = w_n^{(t)} e^{-\beta \cdot \mathbf{1}[y^{(t)} \neq h_n(\boldsymbol{x}^{(t)})]}, \forall n$ ▷ Exponential weight update
8: **end for**

References

[1] Lagrange multiplier. Lagrange multiplier — Wikipedia, the free encyclopedia. [Online; accessed 19-February-2022].

[2] T. L. Maximum. What is a probability simplex? [Online; accessed 19-February-2022].

[3] Simplex. Simplex — Wikipedia, the free encyclopedia. [Online; accessed 19-February-2022].

[4] P. Talwalkar. Viviani's theorem, 2013. [Online; accessed 19-February-2022].

3 Appendix

3.1 Solving the optimization

$$\begin{aligned}
\frac{\delta}{\delta w}\{\eta\frac{1}{2}||w-w^{(t)}||^2 + (f(w^{(t)}) + < w-w^{(t)}, \nabla f(w^{(t)}) >)\} &= 0 \\
\frac{1}{2}(2w+0-2w^{(t)}) + \eta(0+\nabla f(w^{(t)})-0) &= 0 \\
w - w^{(t)} + \eta\nabla f(w^{(t)}) &= 0 \\
w &= w^{(t)} - \eta\nabla f(w^{(t)})
\end{aligned}$$

3.2 Solving the convex conjugate

$$\begin{aligned}
-\psi^*(\theta) &= \frac{1}{2}||w||_2^2 - < \theta, w > \\
&= \frac{1}{2}||\theta||_2^2 - < \theta, \theta > \\
&= -\frac{1}{2}||\theta||_2^2
\end{aligned} \tag{9}$$

3.3 Algebra used in regret bound proof

$$\begin{aligned}
R(u) &\leq \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T}\frac{1}{2\eta}||\theta^{(t+1)}||_2^2 - \frac{1}{2\eta}||\theta^{(t)}||_2^2 - \nabla\frac{1}{2\eta}||\theta^{(t)}||_2^2(\theta^{(t+1)}-\theta^{(t)}) \\
&= \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T}\frac{1}{2\eta}||\theta^{(t+1)}||_2^2 - \frac{1}{2\eta}||\theta^{(t)}||_2^2 - \frac{1}{\eta}\theta^{(t)}(\theta^{(t+1)}-\theta^{(t)}) \\
&= \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T}||\theta^{(t+1)}-\theta^{(t)}||_2^2
\end{aligned} \tag{10}$$

3.4 More Algebra used in regret bound proof

$$\begin{aligned}
R(u) &\leq \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T}||\theta^{(t)} - \eta z^{(t)} - \theta^{(t)}||_2^2 \\
&= \frac{1}{2\eta}||u||_2^2 - \frac{1}{2\eta}||w^{(1)}||_2^2 + \sum_{t=1}^{T}||-\eta z^{(t)}||_2^2 \\
&\leq \frac{1}{2\eta}||u||_2^2 + \sum_{t=1}^{T}\frac{\eta}{2}||z^{(t)}||_2^2
\end{aligned} \tag{11}$$

3.5 Solving for optimal η

$$\begin{aligned}
\frac{\delta}{\delta\eta}\frac{D^2}{2\eta} + \frac{\eta}{2}TG^2 &= 0 \\
\frac{-D^2}{2\eta^2} + \frac{TG^2}{2} &= 0 \\
\frac{-D^2}{2} + \frac{\eta^2 TG^2}{2} &= 0 \\
\eta^2 &= \frac{D^2}{TG^2} \\
\eta &= \frac{D}{G\sqrt{T}}
\end{aligned}$$

3.6 Lagrange multiplier

Lagrange multipliers is a strategy for finding the local maxima and minima of a function ($f(x)$) subject to equality constraints ($g(x)$). [1] It can convert this hard constraint into a term in the function, such that the derivative test of an unconstrained problem can still be applied. The general equation of this method is:

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

Here we show a simple example involving single constraint to help understand this method:

We want to find minima of $f(x, y) = x^2 y$ with constraint $x^2 + y^2 = 1$
Then, the constraint term can be represented as:

$$g(x, y) = x^2 + y^2 - 1$$

And the Lagrangian expression is:

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda g(x, y) = x^y + \lambda(x^2 + y^2 - 1)$$

Set the partial derivative of this equation to 0 and get an equation set:

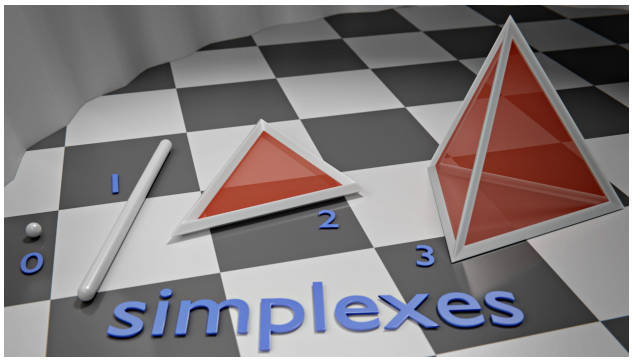
$$\begin{cases} 2xy + 2\lambda x = 0 \\ x^2 + 2xy = 0 \\ x^2 + y^2 - 1 = 0 \end{cases}$$

The minima we're looking for is one of the solutions of this equation set.

3.7 Simplex

In geometry, a simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. The simplex is so-named because it represents the simplest possible polytope in any given space. [3]
A k-simplex is a k-dimensional polytope which is the convex hull of its k + 1 vertices (0-faces). Specifically:

- a 0-simplex is a point
- a 1-simplex is a line segment
- a 2-simplex is a triangle
- a 3-simplex is a tetrahedron



[3]

However, when we talk about probability simplex, it is a mathematical space where each point represents a probability distribution between a finite number of mutually exclusive events. [2]. "Mutually exclusive" means a point on a probability simplex can be represented by K non-negative numbers that add up to 1. Geometrically, it is the $\mathbf{k-1}$ **dimensional** simplex whose vertices are the k standard unit vectors, because the requirement that the numbers sum to 1 reduces the dimensionality by 1.